

# IDENTIFICAÇÃO E CONTROLE EM ROBÔS MÓVEIS UTILIZANDO KIT ROBÓTICO LEGO MINDSTORMS

KLEBER MARCIO DE SOUZA\*, EDNO GENTILHO JUNIOR\*, RUBERLEI GAINO\*, MÁRCIO ROBERTO COVACIC\*

\*Rodovia Celso Garcia Cid (PR 445), km 380, 86057-970  
Centro de Tecnologia e Urbanismo - Universidade Estadual de Londrina  
Londrina, Paraná, Brazil

Emails: klemarcio@gmail.com, edno.gentilho@hotmail.com, ruberlei.gaino@gmail.com, marciocovacic@uel.br

**Abstract**— The aim of this study is to identify and develop a distance control for Lego Mindstorms NXT robot. For this, it requires a model that relates the engine power and distance of a moving object. Based on this principle, an identification system is designed for obtaining a mathematical model that represents this robot. For this, the Toolbox Identification Systems that integrates the software Matlab is used, this tool is as useful building mathematical models of dynamic systems. The mathematical model is used to develop a PID controller that uses as a criterion of the stability Routh-Hurwitz method. For real tests of this control is used kit-robotic Lego Mindstorms NXT, which contains a programmable logic controller that is connected to several components: ultrasonic sensor, sound sensor, touch sensor and servo motors.

**Keywords**— Identification System, stability, kit-robotic Lego

**Resumo**— O objetivo deste estudo é identificar e desenvolver um controle de distância para o robô *Lego mindstorms NXT*. Para isso, é necessário um modelo que relacione a potência do motor e a distância de um objeto em movimento. Com base nesse princípio, uma identificação do sistema será desenvolvida para a obtenção de um modelo matemático que represente esse robô. Para isso, a caixa de Ferramentas Identificação de Sistemas (System Identification Toolbox), que integra o software Matlab é utilizada, essa ferramenta tem como utilidade construir modelos matemáticos de sistemas dinâmicos. O modelo matemático desenvolvido, é usado para o desenvolvimento de um controlador PID que usa como critério de estabilidade o método de Routh-Hurwitz. Para testes reais desse controle, é utilizado o Kit-robótico *Lego Mindstorms NXT*, que contém um controlador lógico programável que é conectado a vários componentes: Sensor ultrasônico, sensor de som, sensor de toque e servo-motores.

**Palavras-chave**— Identificação de sistemas, estabilidade, Kit-Robótico Lego

## 1 Introdução

A área de conhecimento que se dedica a construção de modelos matemáticos a partir de dados produzidos por sistemas dinâmicos é conhecida como identificação de sistemas. Sistemas dinâmicos são aqueles providos de memória, pelo fato de que os sistemas presentes dependem dos estados passados (Aguirre, 2004). Em muitas áreas da ciência, um modelo matemático é importante. Engenheiros e pesquisadores usam modelos matemáticos para projetar sistemas dinâmicos com o intuito de simular e controlar fenômenos da vida real. Assim torna-se necessário conhecer técnicas de modelagem matemáticas para resolver problemas em muitas áreas como aviação e controle. Basicamente, existem duas maneiras de se construir modelos matemáticos (Soderstrom and Stoica, 1989):

- Modelagem matemática: É quando as relações físicas e matemáticas do fenômeno envolvido são conhecidas, e esse conhecimento é utilizado para construção do modelo.
- Identificação de sistemas: Nesse modo, não existe conhecimento matemático do fenômeno, o que existe de fato são dados observados dos fenômenos que servem para construir o modelo.

Esse artigo aborda a definição de um modelo matemático que represente o robô *Lego Mindstorms NXT*, usando o modelo ARX (em inglês, *Autoregressive with Exogenous input*) para identificação do sistema. Esse robô é uma ferramenta de pesquisa e ensino muito utilizada atualmente (Weinberg and Yu, 2003) (Siqueira et al., 2012) e (Pio et al., 2006) e, é equipado com um controlador lógico programável que é conectado a vários componentes eletrônicos: sensor ultrasônico, sensor de som, sensor de toque e servo-motores. O modelo obtido é representado através de função de transferência. Para a identificação do robô, é necessário captar dados que relacionem a potência do motor e a distância de um objeto. Baseando-se nas informações adquiridas desse modelo, é desenvolvido um controlador proporcional-integral-derivativo (PID) que controla a distância do robô em relação a um objeto em movimento e, para obter os ganhos PID utiliza-se de um algoritmo que se baseia no polinômio característico e no critério de estabilidade de Routh-Hurwitz.

## 2 Representação do modelo ARX

Auto-regressão é uma representação matemática do comportamento do processo através de um

modelo que pode ser utilizado para prever o seu comportamento futuro da saída  $y(k)$  em relação a entrada  $u(k)$ . O modelo geral segundo (Aguirre, 2004) é dado por:

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k), \quad (1)$$

cujos polinômios são definidos como:

$$A(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad (2)$$

$$B(q) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad (3)$$

$$C(q) = 1 + c_1q^{-1} + \dots + c_{n_c}q^{-n_c} \quad (4)$$

$$D(q) = 1 + d_1q^{-1} + \dots + d_{n_d}q^{-n_d} \quad (5)$$

$$F(q) = 1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f}, \quad (6)$$

e  $q^{-1}$  representa o operador de atraso, isto é,  $y(k)q^{-1} = y(k-1)$ , e  $e(k)$  um ruído branco.

Pode-se obter ARX a partir da equação (1) fazendo  $C(q) = D(q) = F(q) = 1$ , sendo  $A(q)$  e  $B(q)$  polinômios arbitrários e, considerando que  $A(q)y(k)$  é a parte regressiva e  $B(q)u(k)$  a entrada externa. Obtém-se assim a equação abaixo:

$$A(q)y(k) = B(q)u(k) + e(k). \quad (7)$$

Aparecendo o erro  $e(k)$  diretamente na equação o modelo ARX pode ser classificado como um modelo de erro da equação (Aguirre, 2004). A equação (7) pode ser reescrita da seguinte forma:

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}e(k) \quad (8)$$

a representação ARX pode ser representada na Figura 1.

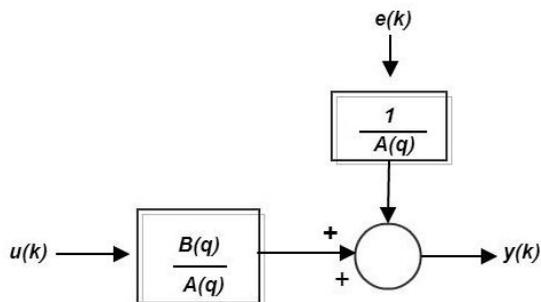


Figura 1: Representação do modelo ARX.

Dado o polinômio representado na Figura 1,

$$A(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a},$$

sendo  $n_a$  o número de parâmetros de  $A(q)$ ,

$$B(q) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b},$$

sendo  $n_b$  o número de parâmetros de  $B(q)$ . Em alguns sistemas existe um atraso de  $u$  a  $y$  considerado como  $n_k$  (Ljung, 1987). A estrutura ARX( $n_a, n_b, n_k$ ) tem o seguinte significado:  $n_a$  parâmetros no denominador (os pólos),  $n_b$  número de parâmetros no numerador (os zeros) e  $n_k$  o atraso das amostras.

### 3 Identificação da planta

Para (Aguirre, 2007), a identificação consiste em algumas etapas iterativas, até que se alcance um modelo adequado: Coleta de informações, escolha da estrutura do modelo, estimação e validação do modelo.

#### 3.1 Coletas de Dados

Para a identificação dinâmica do processo, foram coletados dados referentes à potência aplicada nos motores do robô e à distância percorrida pelo mesmo. Assim, foi aplicada aos motores do robô *NXT* uma potência fazendo o robô locomover-se e, simultaneamente coletar dados através do sensor ultrassônico da distância percorrida, que posteriormente foram gravados na memória do robô. Esses dados foram retirados da memória do robô usando-se a ferramenta *Labview*, que contém uma interface para captura de dados para robôs *Lego Mindstorms NXT*. A Figura 2 mostra a ferramenta de captura, e os primeiros dados coletados do processo.

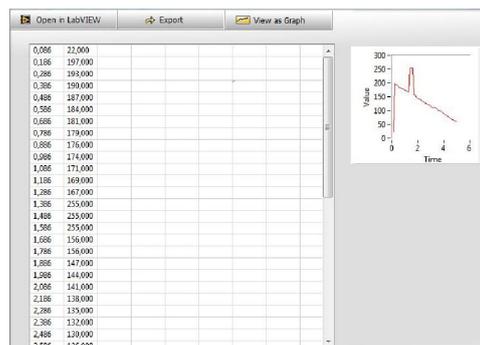


Figura 2: Dados adquiridos pelo Labview

#### 3.2 Obtenção do modelo e estimação

Na identificação, a escolha de uma estrutura que representa o comportamento do sistema dinâmico desejado é importante. Então primeiramente, define-se se a representação é linear ou não-linear (Coelho and dos Santos Coelho, 2004). No trabalho em questão é escolhida uma representação do tipo linear ARX.

Com os dois dados obtidos na coleta de dados, poderemos obter uma estrutura do modelo do robô usando a Toolbox de identificação usada juntamente com o software *Matlab*. Essa ferramenta constrói modelos matemáticos de sistemas dinâmicos utilizando os dados obtidos da entrada e saída do sistema (Ljung, 2007). Fazendo isso, pode-se extrair a função transferência do sistema.

Nessa etapa, foi aplicado aos dados a estrutura escolhida (ARX), para que se possa determinar o melhor parâmetro para minimizar o erro entre a saída do modelo e a saída real. Para a devida

estimação do modelo, foi utilizado o estimador de mínimos quadrados. O modelo representado pela equação (9) foi aproximado para um de primeira ordem ARX(111), com 1 pólo, 1 zero e atraso 1. Utilizando a toolbox de identificação do Matlab obtivemos os seguintes parâmetros:

$$A(q)y(t) = B(q)u(t) + e(t)$$

$$A(q) = 1 - q^{-1}$$

$$B(q) = -0.03235q^{-1}$$

Utilizando-se a função do Matlab *d2c*, transforma-se o modelo ARX que é discreto para um modelo contínuo mostrando a sua função de transferência conforme equação (9).

$$G(s) = \frac{0.5989s + 14.97}{s + 0.001097} \quad (9)$$

### 3.3 Validação

Para avaliar a qualidade de um modelo é necessário verificar o resultado que ele consegue reproduzir para um conjunto de dados diferente dos usados para a obtenção do modelo, porém obtido do mesmo processo de observação (Ljung, 1987). Então, a validação ajuda a avaliar a capacidade de generalização do modelo, ou seja, se o modelo consegue captar a relação entre os dados de entrada e saída (Aguirre, 2004). Com ajuda da toolbox de identificação do Matlab e usando dados diferentes para a validação, obteve-se um modelo semelhante ao obtido na seção 3.2 com 95,4% de coerência com o sistema real. A figura 3 ilustra o modelo obtido em relação ao sistema real.

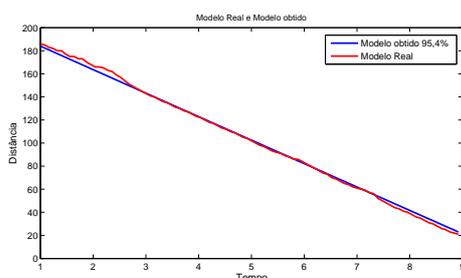


Figura 3: Comparação do modelo real e o obtido

## 4 Sistema de malha fechada com controladores PID

O objetivo de um sistema de controle é obter uma saída desejada com um desempenho desejado. Então, um sistema de controle de malha fechada ou sistema com retroação é um sistema que se utiliza da resposta real, para compará-la com a resposta desejada (Ogata, 2003).

Controladores PID são uma estrutura de controle usada mundialmente em controle industrial.

Esses controladores têm como característica a simplicidade e um desempenho robusto no controle de muitas aplicações (Goodwin et al., 2001). A estrutura PID é composto por três coeficientes: *Proporcional*, *Integral*, *derivativo*, que podem ser variados a fim de obter a resposta ideal (Dorf and Bishop, 2001). Considere a malha fechada do sistema com uma entrada e uma saída (SISO):

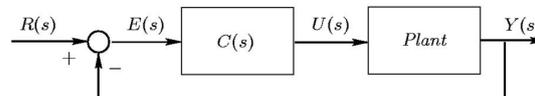


Figura 4: Malha fechada - SISO

As expressões para os controladores PI, PD, e PID, podem ser descritas pela relação da saída do controlador  $U(s)$  e o sinal de erro  $E(s)=R(s)-Y(s)$  (Ogata, 2003). A equação de um controlador PID pode ser escrita da seguinte forma :

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}, \quad (10)$$

e a sua função de transferência é:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right),$$

onde  $K_p$  é o ganho proporcional,  $T_i$  é o tempo integrativo e  $T_d$  é o tempo derivativo. Como observado na fórmula (10) os membros dessa estrutura são combinados formando três modos de controle: *Proporcional(P)*, *Integral(I)*, *Derivativo(D)*. Essa combinação tem a vantagem de usufruir das características individuais de cada uma. Na próxima seção, será discutida a obtenção de valores para  $K_p$ ,  $T_i$  e  $T_d$ , a fim de garantir estabilidade no robô Lego Mindstorms NXT.

## 5 Estabilidade do Robô

Ao se projetar um sistema de controle em malha fechada, a estabilidade torna-se um assunto de muita importância. Um sistema instável não tem aplicação prática. A estabilidade de um sistema pode ser definida em termos da localização dos pólos no plano  $s$ . Plano  $s$  é um plano cartesiano que é usado para representar números complexos geometricamente e, pode ser dividido em: semi-plano direito (SPD) e semi-plano esquerdo (SPE). O eixo horizontal representa a parte real ( $\sigma$ ) e o eixo vertical representa a parte imaginária ( $j\omega$ ) (Dorf and Bishop, 2001). Para que um sistema seja estável, todos os pólos da função de transferência desse sistema devem estar no SPE. Com a função de transferência obtida na Seção 3.2, pode-se obter uma região de estabilidade, para desenvolver um controle para o robô Lego, que é objetivo desse trabalho.

A região de estabilidade é obtida utilizando-se o método de Routh-Hurwitz. Esse método indica

a quantidade de pólos no SPD, no SPE e sobre eixo  $iw$ , sem a necessidade de calcular os pólos. Segundo (Nise, 2010) o método de Routh requer dois passos:

1. Gerar o arranjo ou tabela de Routh-Hurwitz.
2. Interpretar os valores obtidos, identificando a localização dos pólos no SPD, SPE e  $(iw)$ .

Com base na equação do polinômio característico (11), segue a definição para montagem do arranjo de Routh. Considere o seguinte polinômio:

$$D(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0 \quad (11)$$

Desenvolvendo o arranjo fica:

$a_n$	$a_{n-2}$	$a_{n-4}$	...
$a_{n-1}$	$a_{n-3}$	$a_{n-5}$	...
$b_1$	$b_2$	$b_3$	...
$c_1$	$c_2$	$c_3$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Tabela 1: Tabela de Routh

onde os elementos  $b_i$  e  $c_i$  podem ser calculados da seguinte maneira:

$$b_i = \frac{a_{n-1} \cdot a_{n-2i} - a_n \cdot a_{n-2i-1}}{a_{n-1}}$$

$$c_i = \frac{b_1 \cdot a_{n-2i-1} - a_{n-1} \cdot b_i + 1}{b_1}$$

Para determinar a região de estabilidade para os ganhos do controlador PID, mostrado na figura 4 utilizou-se um algoritmo que se baseia no polinômio característico e é descrito em (Teixeira et al., 2007). Nesse algoritmo digita-se os coeficientes do polinômio característico e obtêm-se valores para os parâmetros P, I, D respectivamente. A execução do algoritmo se dá digitando os coeficientes da função de transferência do numerador e denominador.

No caso do robô, sua função de transferência é dada por (9). O polinômio do numerador é  $0.5989s + 14.97$  e do denominador é  $s + 0.001097$ . Esses devem ser digitados da seguinte forma respectivamente  $[0.5989 \ 14.97]$  e  $[1 \ 0.001097]$ . Após isso, escolhe-se mais duas especificações: taxa de decaimento ou estabilidade. Por fim, escolhe-se o tipo de controlador usado. Para o caso da escolha do PID, valores finais e iniciais devem ser digitados para  $K_d$  e  $K_i$  respectivamente. Segue o resultado da execução do algoritmo em questão:

Type the coefficients of N(s): [0.5989 14.97]  
Type the coefficients of D(s): [1 0.001097]

Type "1" if you want to specify a decay rate or "0" if you want to see the stability range: 0

Choose the type of the controller:

- 1 - Proportional (P);
- 2 - Integral (I);
- 3 - Derivative (D);
- 4 - Proportional-Integral (PI);
- 5 - Proportional-Derivative (PD);
- 6 - Integral-Derivative (ID);
- 7 - Proportional-Integral-Derivative (PID);
- 0 - Cancel.

Option: 7

Default tolerance is  $10^{-6}$ .

Is this value suitable for you (Y/N)? y

Type the initial value of kd: 5

Type the final value of kd: 10

Type the step of kd: 1

Type the initial value of ki: 5

Type the final value of ki: 10

Type the step of ki: 1

Após da digitação dos coeficientes obtêm-se a faixa de valores para P, I, D. Alguns resultados obtidos são mostrados.

For kd = 5 and ki = 5:	Solution: kp > -0.0027063
For kd = 5 and ki = 6:	Solution: kp > -0.0032319
For kd = 5 and ki = 7:	Solution: kp > -0.0037572
For kd = 5 and ki = 8:	Solution: kp > -0.0042822
$\vdots$	$\vdots$

Tabela 2: Faixa de valores para o PID

## 6 Resultados e discussão

Os resultados obtidos mostram que o sistema pode ser representado por um modelo de baixa ordem (ARX111). Após a identificação, é testado o controle real no robô. Respectivamente, inserimos os ganhos para P = 10, I = 8, D = 5 adquiridos na seção 5 e tabela 2 no algoritmo. Assim, o conjunto de valores inseridos no algoritmo, está dentro da região de estabilidade. Por fim, o algoritmo é carregado para a memória do robô.

Para se implementar um controle PID discreto no Lego Mindstorms, usou-se a linguagem chamada NXC, que é criada especialmente para construir aplicações para Lego Mindstorms e, que é suportada por um ambiente de programação (IDE) chamado *Bricx Command Center (Bricx Command Center 3.3, 2013)*.

Estipulando para que o robô permanecesse a uma distância de 21cm de um objeto, o projeto de identificação e estabilidade do robô mostrou-se eficiente em testes reais, mantendo-se na distância estipulada a partir do objeto em movimento. Com base em dados reais coletados da memória do robô a execução em questão pode ser vista graficamente na Figura 5. Nessa figura, observa-se uma análise livre do modelo, na qual digita-se valores

quaisquer para P, I, D. Obtendo assim, um sistema instável. E, em outro teste usando os valores obtidos do modelo, mostra a estabilização do sistema na distância pretendida.

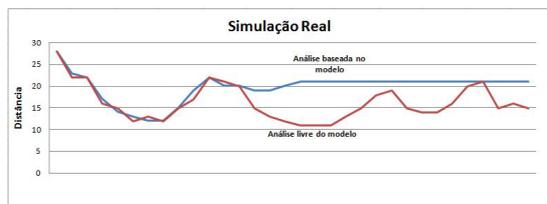


Figura 5: Simulação do controlador

## 7 Conclusão

Em diversas situações, sistemas computacionais são necessários para o entendimento e funcionamento de sistemas dinâmicos. Esse trabalho obteve um modelo matemático do robô Lego mindstorms NXT; para isso, técnicas de identificação de sistemas e controle foi utilizado. Para aplicação dessas técnicas a toolbox de identificação do Matlab se mostrou de grande utilidade nas aplicações em Modelagem e Identificação de sistemas dinâmicos. Em testes de validação, o modelo de primeira ordem ARX(111) gerado apresentou uma boa validação obtendo uma coerência de 95,4% com o modelo real. Após o desenvolvimento do modelo matemático, procurou-se adquirir parâmetros para estabilizar o robô em questão a uma distância pré-definida de um objeto em movimento. Assim, fazendo-se uso do critério de Routh-Hurwitz, parâmetros para P, I, D foram conseguidos com base no polinômio característico do sistema e inseridos no algoritmo de controle do robô. Em teste reais, conclui-se então que o controle desenvolvido pode ser utilizado no robô com um grau de liberdade. Os resultados obtidos com esse trabalho, abrem caminho para outras aplicações feitas em robótica, tais como controle de robô com mais de um grau de liberdade e desenvolvimento de controle de distância para sistemas automobilísticos.

## Agradecimentos

- Universidade Estadual de Londrina - Programa de pós-graduação em Engenharia Elétrica.
- Amigos e orientadores que me ajudaram direta e indiretamente nesse trabalho.

## Referências

Aguirre, L. (2007). *Enciclopédia de automática: Controle e automação volume III*, Blucher.

Aguirre, L. A. (2004). *Introdução à identificação de sistemas—Técnicas lineares e não-lineares aplicadas a sistemas reais*, editora UFMG.

*Brix Command Center 3.3* (2013). <http://bricxcc.sourceforge.net/>. Acessado em 20/05/2013.

Coelho, A. A. R. and dos Santos Coelho, L. (2004). *Identificação de sistemas dinâmicos lineares*, Didáctica (Editora da UFSC), Editora da UFSC.

Dorf, R. C. and Bishop, R. H. (2001). *Sistemas de Controle Modernos, 8 edição*, LTC.

Goodwin, G. C., Graebe, S. F. and Salgado, M. E. (2001). *Control system design*, Vol. 240, Prentice Hall Upper Saddle River.

Ljung, L. (1987). *System identification: theory for the user*, Prentice-Hall information and system sciences series, Prentice-Hall.

Ljung, L. (2007). *System Identification Toolbox 7: User's Guide*, MathWorks, Incorporated.

Nise, N. S. (2010). *Control Systems Engineering, 6th Edition*, Wiley.

Ogata, K. (2003). *Engenharia de controle moderno*, Pearson Prentice Hall.

Pio, J. L. d. S., Castro, T. H. C. d. and de Castro Júnior, A. N. (2006). A robótica móvel como instrumento de apoio à aprendizagem de computação, *Anais do Simpósio Brasileiro de Informática na Educação*, Vol. 1, pp. 497–506.

Siqueira, M. D., Conforto, J. and Vallim, M. B. (2012). An FPGA-based robotics platform for educational and research applications, *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, IEEE, pp. 313–318.

Soderstrom, T. and Stoica, P. (1989). *System Identification*, Prentice Hall International Series In Systems And Control Engineering, Prentice Hall.

Teixeira, M. C., Assunção, E. and Covacic, M. R. (2007). Proportional controllers: direct method for stability analysis and matlab implementation, *Education, IEEE Transactions on* **50**(1): 74–78.

Weinberg, J. B. and Yu, X. (2003). Robotics in education: Low-cost platforms for teaching integrated systems, *Robotics & Automation Magazine, IEEE* **10**(2): 4–6.